Multi-turn Consistent Image Editing
Supplementary Material

Contents

A. Related Work

Image Inversion: Image inversion transforms a clean image into a latent Gaussian noise representation. Sampling from this representation enables controlled image editing and reconstruction. Diffusion-based inversion originated with DDPM [15, 28], which progressively add noise to an image. DDIM [32], a deterministic variant of DDPMs, allows for significantly faster inversion. However, early inversion techniques often lacked sufficient accuracy. Null-text inversion [26] addresses this limitation by optimizing a null-text embedding, effectively leveraging the inherent bias of the inversion process. Negative-Prompt-Inversion [25] mathematically derives the optimization process of null-text inversion, thereby accelerating the inversion process. Direct-Inversion [18] incorporates the inverted noise corresponding to each timestep within the denoising process to mitigate content leakage.

Image Editing: To maintain the consistency of edited images with the source image, several approaches constrain the editing results. One strategy, employed by [33, 38, 39], involves tuning additional parameters to inject source image information or providing structural control through masks, canny edges, or depth maps. Another prominent approach, stemming from Prompt2Prompt [13], manipulates attention maps to preserve image structure, as seen in various editing methods [3–6, 9, 12, 14, 19, 22, 27, 34, 36, 40]. Furthermore, mask-based techniques have proven effective in enhancing both preservation and editability. For instance, [3, 7, 16, 24] utilize automatically generated masks for more accurate text-guided image generation. Flow-based image editing methods [1, 2, 11, 21, 23] have demonstrated strong performance in single-turn editing. Building upon this foundation, RF-Inversion [31] employs a single-objective LQR control framework. FireFlow [10] and RF-Solver [35] further refine the process by focusing on reducing single-step simulation error through second-order ODE solvers. Our work addresses the specific challenges of multi-turn editing. We leverage second-order ODEs for accurate single-step inversion and, crucially, introduce a dual-objective LQR and adaptive attention guidance to maintain coherence and control across multiple editing steps.

Joseph et al. [17] explored techniques for manipulating images directly within the latent space. ChatEdit [8] and TextBind [20] leverage the summarization capabilities of LLMs to streamline editing workflows. Similarly, Yang et al. [37] employ a self-refinement strategy using GPT-4V [29] to support interactive image editing. While these methods enhance editing efficiency, they underutilize the image generation model's full potential. In contrast, our work focuses on multi-turn image editing by directly optimizing the image generation model's capabilities, enabling consistent edits across multiple iterations without relying on external language models.

B. Datasets

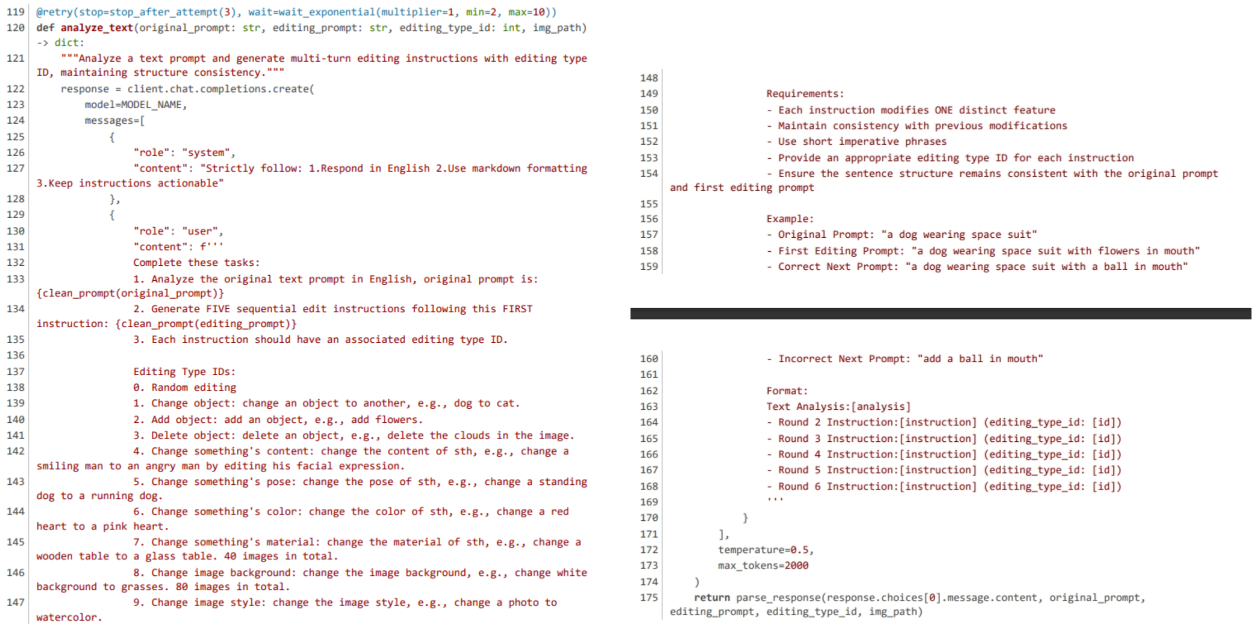
Since there are no existing datasets for multi-turn image editing, we propose an extended dataset based on PIE-Bench [18] to facilitate evaluation. This extension allows for testing multi-turn editing while maintaining alignment with existing single-turn editing benchmarks. PIE-Bench consists of 10 editing types, as outlined below:

1. Random editing: Random prompt written by volunteers or examples in previous research.
2. Change object: Change an object to another, e.g., dog to cat.
3. Add object: add an object, e.g., add flowers.
4. Delete object: delete an object, e.g., delete the clouds in the image.
5. Change sth's content: dhange the content of sth, e.g., change a smiling man to an angry man by editing his facial expression.
6. Change sth's pose: dhange the pose of sth, e.g., change a standing dog to a running dog.
7. Change sth's color: change the color of sth, e.g., change a red heart to a pink heart.
8. Change sth's material: change the material of sth, e.g., change a wooden table to a glass table.
9. Change image background: change the image background, e.g., change white background to grasses.
10. Change image style: change the image style, e.g., change a photo to watercolor.

PIE-Bench is a dataset designed for single-turn editing, where each image is paired with an original prompt and an editing instruction. To extend it for multi-turn editing, we utilize OpenAI's GPT-4 Turbo to generate additional editing instructions. Based on the original prompt and the first-round editing instruction, we randomly select one of the ten editing types and generate five additional rounds of editing instructions for each image. The prompts used for generating editing instructions are shown in Fig. 1.

C. Technical Proofs

This section provides detailed technical proofs for the theoretical results discussed in this paper.



```
@retry(stop=stop_after_attempt(3), wait=wait_exponential(multiplier=1, min=2, max=10))
def analyze_text(original_prompt: str, editing_prompt: str, editing_type_id: int, img_path) -> dict:
    """Analyze a text prompt and generate multi-turn editing instructions with editing type ID, maintaining structure consistency."""
    response = client.chat.completions.create(
        model=MODEL_NAME,
        messages=[
            {
                "role": "system",
                "content": "Strictly follow: 1.Respond in English 2.Use markdown formatting 3.Keep instructions actionable"
            },
            {
                "role": "user",
                "content": f'''
                Complete these tasks:
                1. Analyze the original text prompt in English, original prompt is: {clean_prompt(original_prompt)}
                2. Generate FIVE sequential edit instructions following this FIRST instruction: {clean_prompt(editing_prompt)}
                3. Each instruction should have an associated editing type ID.

                Editing Type IDs:
                0. Random editing
                1. Change object: change an object to another, e.g., dog to cat.
                2. Add object: add an object, e.g., add flowers.
                3. Delete object: delete an object, e.g., delete the clouds in the image.
                4. Change something's content: change the content of sth, e.g., change a smiling man to an angry man by editing his facial expression.
                5. Change something's pose: change the pose of sth, e.g., change a standing dog to a running dog.
                6. Change something's color: change the color of sth, e.g., change a red heart to a pink heart.
                7. Change something's material: change the material of sth, e.g., change a wooden table to a glass table. 40 images in total.
                8. Change image background: change the image background, e.g., change white background to grasses. 80 images in total.
                9. Change image style: change the image style, e.g., change a photo to watercolor.

                Requirements:
                - Each instruction modifies ONE distinct feature
                - Maintain consistency with previous modifications
                - Use short imperative phrases
                - Provide an appropriate editing type ID for each instruction
                - Ensure the sentence structure remains consistent with the original prompt and first editing prompt

                Example:
                - Original Prompt: "a dog wearing space suit"
                - First Editing Prompt: "a dog wearing space suit with flowers in mouth"
                - Correct Next Prompt: "a dog wearing space suit with a ball in mouth"
```

```
                - Incorrect Next Prompt: "add a ball in mouth"

                Format:
                Text Analysis:[analysis]
                - Round 2 Instruction:[instruction] (editing_type_id: [id])
                - Round 3 Instruction:[instruction] (editing_type_id: [id])
                - Round 4 Instruction:[instruction] (editing_type_id: [id])
                - Round 5 Instruction:[instruction] (editing_type_id: [id])
                - Round 6 Instruction:[instruction] (editing_type_id: [id])
                '''
            }
        ],
        temperature=0.5,
        max_tokens=2000
    )
    return parse_response(response.choices[0].message.content, original_prompt, editing_prompt, editing_type_id, img_path)
```

Figure 1. Prompts for GPT4-Turbo genrating multi-turn editing instrctions.

C.1. Proof of Proposition 1

Proof. The original problem with a single target is formulated as:

$$V(c) \coloneqq \int_0^1 \frac{1}{2} \|c\left(Z_t, t\right)\|_2^2 \, \mathrm{d}t + \frac{\lambda}{2} \|Z_1 - X_1\|_2^2$$

The extended problem considering multiple targets is expressed as:

$$V(c) \coloneqq \int_0^1 \frac{1}{2} \|c\left(Z_t, t\right)\|_2^2 \, \mathrm{d}t + \sum_{i=1}^{n} \frac{\lambda_i}{2} \|Z_1 - X_i\|_2^2$$

Rewriting the extended problem, the sum of squared distances is reformulated as:

$$\sum_{i=1}^{n} \frac{\lambda_i}{2} \|Z_1 - Y_i\|_2^2 = \frac{\sum_{i=1}^{n} \lambda_i}{2} \|Z_1 - \mu\|_2^2 + c$$

where μ is defined as the weighted average of the targets:

$$\mu = \frac{\sum_{i=1}^{n} \lambda_i Y_i}{\sum_{i=1}^{n} \lambda_i}$$

and c corresponds to a constant value, which is irrelevant to Z_1. By defining a new target μ and a new weight $\lambda' = \sum_{i=1}^{n} \lambda_i$, the extended problem simplifies to:

$$V(c) \coloneqq \int_0^1 \frac{1}{2} \|c\left(Z_t, t\right)\|_2^2 \, \mathrm{d}t + \frac{\lambda'}{2} \|Z_1 - \mu\|_2^2$$

This formulation is structurally identical to the single object LQR problem, where the target Y_1 is replaced by μ and the weight λ is replaced by λ'. □

C.2. Solution to LQR Problem

The standard approach to solving an LQR problem is the minimum principle theorem that can be found in control literature. We follow this approach and provide the full proof below for completeness. The Hamiltonian of the LQR problem is given by

$$H(\mathbf{z}_t, \mathbf{p}_t, \mathbf{c}_t, t) = \frac{1}{2} \|\mathbf{c}_t\|^2 + \mathbf{p}_t^T \mathbf{c}_t. \tag{1}$$

For $\mathbf{c}_t^* = -\mathbf{p}_t$, the Hamiltonian attains its minimum value: $H(\mathbf{z}_t, \mathbf{p}_t, \mathbf{c}_t^*, t) = -\frac{1}{2} \|\mathbf{p}_t\|^2$. Using the minimum principle theorem, we get

$$\frac{\mathrm{d}\mathbf{p}_t}{\mathrm{d}t} = \nabla_{\mathbf{z}_t} H\left(\mathbf{z}_t, \mathbf{p}_t, \mathbf{c}_t^*, t\right) = 0; \tag{2}$$

$$\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} = \nabla_{\mathbf{p}_t} H\left(\mathbf{z}_t, \mathbf{p}_t, \mathbf{c}_t^*, t\right) = -\mathbf{p}_t; \tag{3}$$

$$\mathbf{z}_0 = \mathbf{y}_0; \tag{4}$$

$$\mathbf{p}_1 = \nabla_{\mathbf{z}_1} \left(\frac{\lambda}{2} \|\mathbf{z}_1 - \mathbf{y}_1\|_2^2 \right) = \lambda \left(\mathbf{z}_1 - \mathbf{y}_1\right). \tag{5}$$

From (2), we know $\mathbf{p}_t$ is a constant $\mathbf{p}$. Using this constant in (3) and integrating from $t \to 1$, we have $\mathbf{z}_1 = \mathbf{z}_t - \mathbf{p}(1-t)$. Substituting $\mathbf{z}_1$ in (4),

$$\mathbf{p} = \lambda(\mathbf{z}_t - \mathbf{p}(1-t) - \mathbf{y}_1) = \lambda(\mathbf{z}_t - \mathbf{y}_1) - \lambda(1-t)\mathbf{p},$$

which simplifies to

$$\begin{aligned} \mathbf{p} &= \left(1 + \lambda(1-t)\right)^{-1} \lambda(\mathbf{z}_t - \mathbf{y}_1) \\ &= \left(\frac{1}{\lambda} + (1-t) \right)^{-1} (\mathbf{z}_t - \mathbf{y}_1). \end{aligned}$$

Taking the limit $\lambda \to \infty$, we get $\mathbf{p} = \frac{\mathbf{z}_t - \mathbf{y}_1}{1-t}$ and the optimal controller $\mathbf{c}_t^* = \frac{\mathbf{y}_1 - \mathbf{z}_t}{1-t}$. Since $u_t(\mathbf{z}_t|\mathbf{y}_1) = \mathbf{y}_1 - \mathbf{y}_0$, the proof follows by substituting $\mathbf{y}_0 = \frac{\mathbf{z}_t - t\mathbf{y}_1}{1-t}$.

In conclusion, the formulation with multiple targets can be regarded as a special case of the original single-target Linear Quadratic Regulator (LQR) problem. In this interpretation, the effective target is a weighted average of the individual targets, and the effective weight is the sum of the individual weights. This allows for the seamless application of the optimal control techniques developed for the single-target scenario to be extended to handle the multi-target problem by treating the weighted average target as the effective target.

D. Pseudo-code for the k-th Turn of Image Editing

To improve reproducibility, the pseudo-code for the k-th turn of image editing is provided in Algorithm 1.

E. Limitations

E.1. Editing Iterations

As shown in Fig. 4, our method effectively preserves the natural appearance of images across multiple editing rounds, whereas other methods exhibit noticeable artifacts. However, due to limitations in dataset generation, we created only five rounds of editing instructions. Additionally, errors from ChatGPT restricted our benchmark evaluation to four editing turns.

As a result, we have not yet fully explored the potential of our method across a larger number of editing iterations. As seen in the reconstruction results presented in this paper, most flow-based inversion methods begin to exhibit significant semantic drift by the fourth reconstruction. In contrast, our multi-turn reconstruction results demonstrate that even after 10 reconstruction steps reconstruction, our method maintains high-quality outputs.

Since our evaluation was limited to only four editing rounds, a comprehensive comparison between methods remains incomplete. Moving forward, we aim to extend the multi-turn dataset to support a greater number of editing iterations for a more thorough evaluation.

Algorithm 1 Multi-turn Editing Denoising ODE

```
Require: Editing turn $k$, discretization steps $N$, target text "prompt", structured noise $X_1^k$, prompt embedding network $\Phi$, Flux model $v(\cdot, \cdot, \cdot; \varphi)$, time steps $t = [t_{N-1}, ..., t_0]$, mask of activated attention map in layer $l$ is $m_t^l$, LQR guidance controller $\eta$, $\lambda$
Ensure: $k$th turn edited image $X_0^k$
Initialize $v_{t_{N-1}} = v(X_{t_{N-1}}^k, t_{N-1}, \Phi(\text{prompt}); \varphi)$
$\Delta t = t_{N-2} - t_{N-1}$
$X_{t_{N-1}+\frac{1}{2}\Delta t}^k = X_{t_{N-1}}^k + \frac{1}{2}\Delta t \cdot v_{t_{N-1}}$
Initialize $v_{t_{N-1}+\frac{1}{2}\Delta t} = v(X_{t_{N-1}+\frac{\Delta t}{2}}^k, t_{N-1+\frac{1}{2}\Delta t}, \Phi(\text{prompt}); \varphi)$    {Run & Save to GPU Memory}
$X_{dual}^k = [(X_0^0 - X_{t_{N-1}+\frac{\Delta t}{2}}^k) + \lambda((X_0^{k-1} - X_{t_{N-1}+\frac{\Delta t}{2}}^k) - (X_0^0 - X_{t_{N-1}+\frac{\Delta t}{2}}^k))]/(1 - t_{N-1})$ {LQR guidance, integration of historical data}
$X_{t_{N-2}}^k = X_{t_{N-1}}^k + \Delta t \cdot [v_{t_{N-1}+\frac{1}{2}\Delta t} + \eta(X_{dual}^k - v_{t_{N-1}+\frac{1}{2}\Delta t})]$
for $i = N - 2 : 0$ do
    $\hat{v}_{t_i} \leftarrow v_{t_{i+1}+\frac{1}{2}\Delta t_{i+1}}$    {Load from GPU Memory & $m_{t_{i-1}}^l$ guide attention's reweight & Run}
    $\Delta t = t_{i-1} - t_i$
    $X_{t_i+\frac{1}{2}\Delta t}^k = X_{t_i}^k + \frac{1}{2}\Delta t \cdot \hat{v}_{t_i}$
    $v_{t_i+\frac{1}{2}\Delta t} = v(X_{t_i+\frac{1}{2}\Delta t}^k, t_i + \frac{1}{2}\Delta t, \Phi(\text{prompt}); \varphi)$    {Run & Save to GPU Memory}
    $X_{dual}^k = [(X_0^0 - X_{t_{i+\frac{1}{2}\Delta t}}^k) + \lambda((X_0^{k-1} - X_{t_{i+\frac{1}{2}\Delta t}}^k) - (X_0^0 - X_{t_{i+\frac{1}{2}\Delta t}}^k))]/(1 - t_i)$    {LQR guidance, integration of historical data}
    $X_{t_{i-1}}^k = X_{t_i}^k + \Delta t \cdot [v_{t_i+\frac{1}{2}\Delta t} + \eta(X_{dual}^k - v_{t_i+\frac{1}{2}\Delta t})]$
end for
return $X_0^k$ =0
```

E.2. First Round Editing

LQR-guided methods are highly effective in aligning distributions, particularly in transforming atypical distributions into typical ones. This capability is essential for maintaining coherence in multi-turn editing. However, in single-turn editing, LQR guidance can disrupt the original flow matching process to some degree. Consequently, the performance of our method in the initial editing round is suboptimal. Future work could explore alternative methods to integrate information across editing iterations.

F. Additional Experiments

In this section, we begin by presenting comprehensive experimental metrics across multiple editing rounds Sec. F.2. Next, we showcase quantitative results demonstrating that our method is highly effective for multi-turn editing, excelling in both editability and structure preservation Sec. F.3. Finally, we conduct additional ablation studies to analyze the functionality of key components.

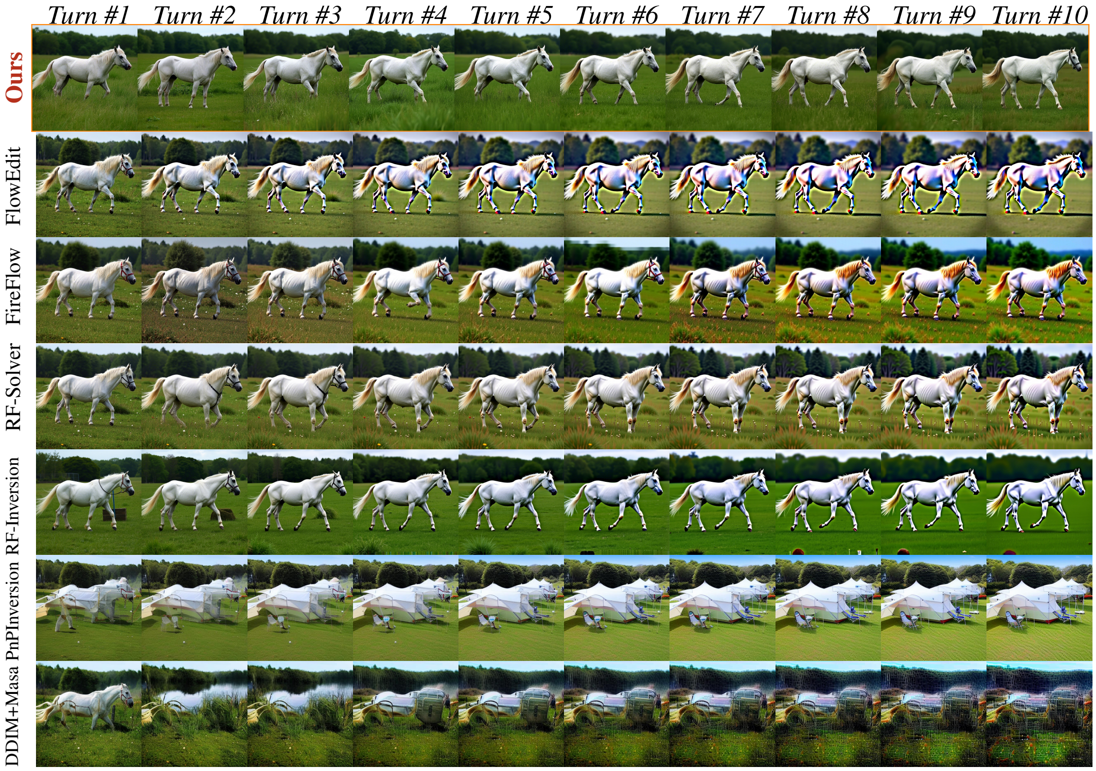
F.1. Multi-turn Reconstruction

To evaluate long-term performance, we include a 10-turn reconstruction example in Sec. F.1, demonstrating that our method remains stable with fewer drift issues.

F.2. Quantitative Results for Multi-turn Editing

We utilize CLIP-T [30] to measure image-text alignment, while CLIP-I and structure-distance assess the similarity between the edited and original images. The FID is employed to evaluate the quality of the generated images. Additionally, since PIE-bench provides a mask labeling the edited area, we use CLIP-Edit to measure image-text similarity specifically within the edited region.

Quantitative results are presented in Tab. 1 and Tab. 2. Our method demonstrates a strong balance between content preservation and editing capability, particularly in the fourth round of editing. Notably, for the FID and structure-distance metrics, our method maintains stable performance across multiple editing turns, whereas most competing methods exhibit a continuous increase in both structure distance and FID as the number of editing rounds grows. Furthermore, our multi-turn approach achieves comparable performance to state-of-the-art flow-based editing methods in the initial rounds and delivers outstanding performance in later rounds. To comprehensively evaluate overall performance, we compare our method with baseline methods on fourth-turn editing results. All metrics are normalized to a 0-10 ranking and visualized using a radar plot, which shows that our method strikes a balance across all metrics.(Fig. 2)



Methods	Round 1			Round 2			Round 3			Round 4		
	FID	Clip-T	Clip-I	FID	Clip-T	Clip-I	FID	Clip-T	Clip-I	FID	Clip-T	Clip-I
Ours	2.554	26.19	0.910	4.015	26.56	0.903	5.115	26.81	0.897	5.553	26.83	0.894
RF-Inv.	1.854	24.41	0.928	3.015	24.09	0.919	4.324	24.06	0.909	5.740	24.10	0.904
StableFlow	1.699	23.94	0.940	5.971	23.98	0.932	12.413	23.94	0.914	20.624	24.23	0.899
FlowEdit	0.998	26.28	0.932	3.706	26.34	0.914	8.405	26.36	0.903	14.547	26.70	0.894
RF-Solver	1.450	25.58	0.931	3.419	25.55	0.922	6.603	25.62	0.912	11.581	25.52	0.906
FireFlow	5.579	27.72	0.891	8.279	27.87	0.883	8.405	27.94	0.878	12.375	28.28	0.873
MasaCtrl	1.647	23.98	0.933	4.518	23.80	0.915	7.609	23.91	0.900	10.811	23.80	0.886
PnPInv.	2.222	25.25	0.915	4.927	25.67	0.901	7.703	25.47	0.889	10.262	25.77	0.872

Table 1. **Quantitative Results of Multi-Turn Editing.** The best results are highlighted in green, while the second-best results are marked in purple. Our method demonstrates a balance between CLIP-I and CLIP-T while achieving the best FID score at the fourth-turn editing.

Methods	Round 1		Round 2		Round 3		Round 4	
	CLIP-edit	Structure	CLIP-Edit	Structure	CLIP-Edit	Structure	CLIP-Edit	Structure
Ours	23.596	0.0475	23.120	0.0587	23.294	0.0652	23.021	0.0580
RF-Inv.	21.573	0.0326	21.834	0.0411	21.920	0.0471	21.945	0.0525
StableFlow	21.187	0.0190	21.581	0.0375	21.926	0.0589	22.051	0.0785
FlowEdit	23.393	0.0289	23.378	0.0493	23.237	0.0668	22.941	0.0813
RF-Solver	22.536	0.0249	23.101	0.0359	23.229	0.0488	22.581	0.0611
FireFlow	24.226	0.0780	23.843	0.1040	23.524	0.1240	23.208	0.1420
MasaCtrl	21.073	0.0271	21.557	0.0456	21.621	0.0595	21.776	0.0709
PnPInv.	22.502	0.0218	22.859	0.0424	22.788	0.0580	22.752	0.0692

Table 2. **Quantitative Results of Multi-Turn Editing.** The best results are highlighted in green, while the second-best results are marked in purple.

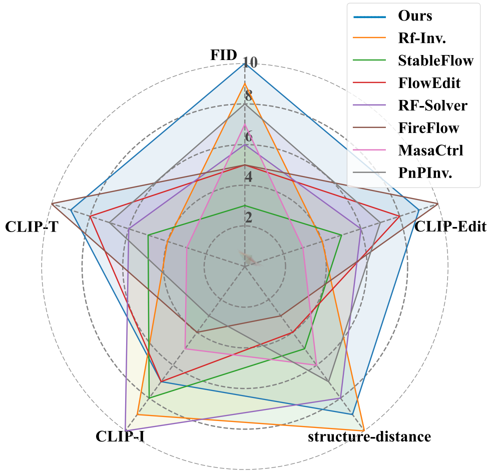


Figure 2. We rank the performance of our method compared to baseline methods in the fourth round of editing. Our method performs well in both text similarity and fidelity to the original image.

F.3. Qualitative Results for Multi-turn Editing

Existing metrics cannot accurately assess image quality. For example, our selected baseline diffusion-based methods produce noticeable artifacts compared to flow-based methods. However, qualitative evaluations do not always capture these differences effectively.

To address this, we conduct additional qualitative experiments on both natural and artificial images. The results for natural image editing are shown in Fig. 3, Fig. 4, while artificial image results are presented in Fig. 5 and Fig. 6. In both categories, our method achieves a high success rate in image editing. Equally important, our edited images consistently preserve key features of the original image across multiple editing steps, including color, lighting and background. This balance between content preservation and editing effectiveness aligns with the quantitative results in Sec. F.2. Artistic paintings are among the most challenging images to edit and reconstruct. We present full experimental results on multi-turn reconstruction in this domain, showing that our method can effectively complete the editing process.

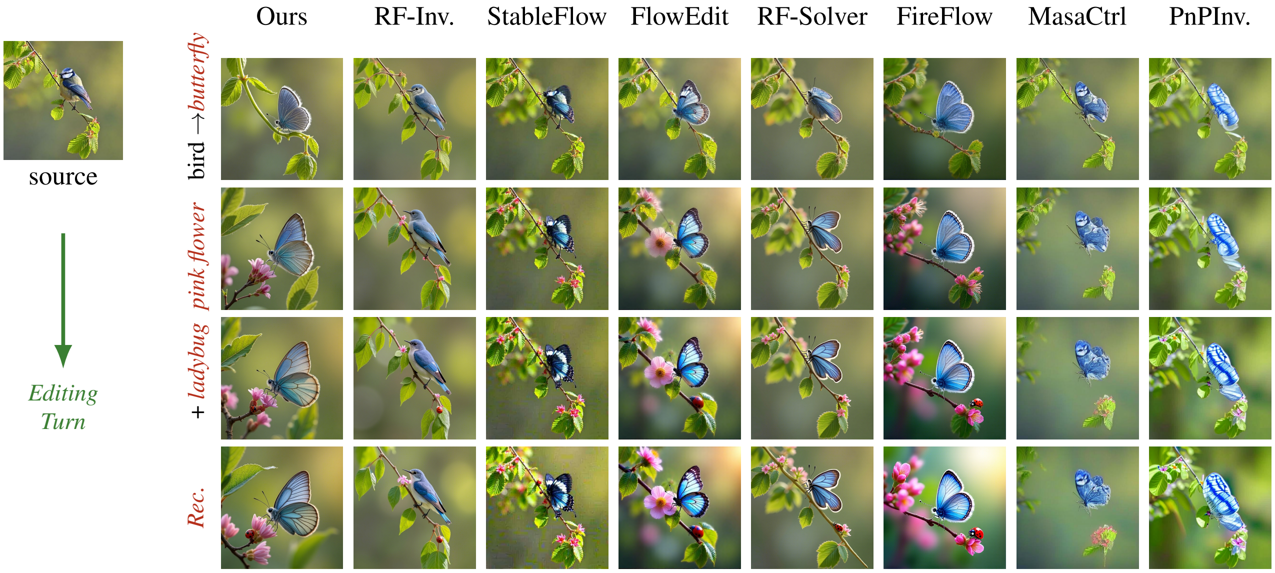


Figure 3. Our method consistently follows the color tone of the original image while achieving the desired editing. The second prompt is "sitting on a pink flower", while the third prompt is "with a red ladybug".

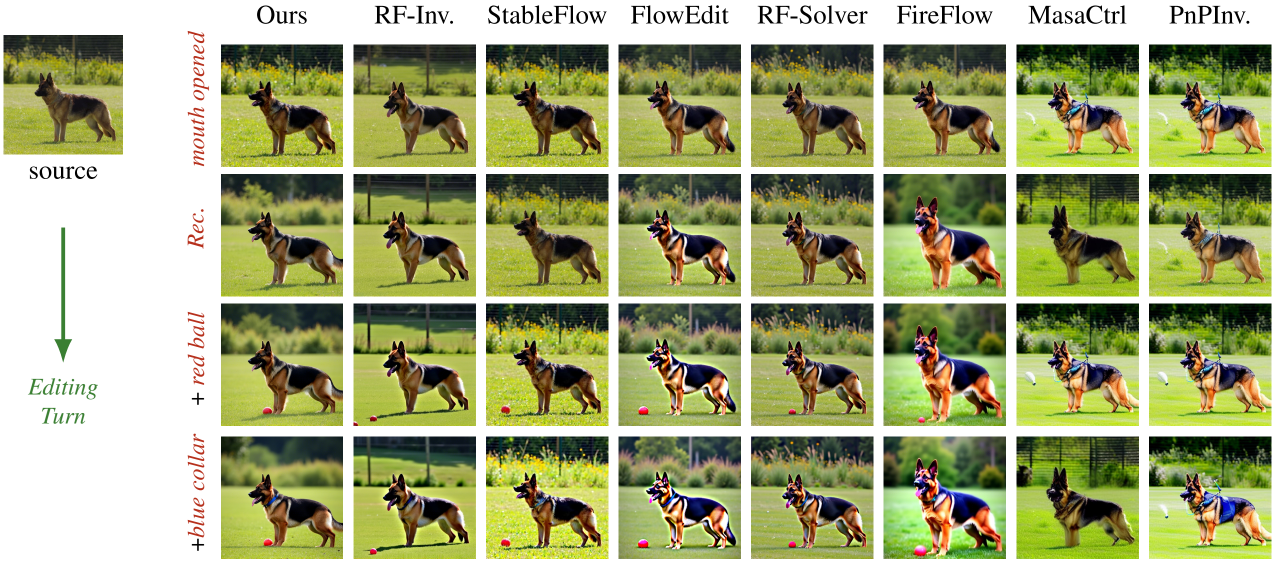


Figure 4. Quantitative Results on Natural Animals. Our method successfully performs edits without introducing artifacts.

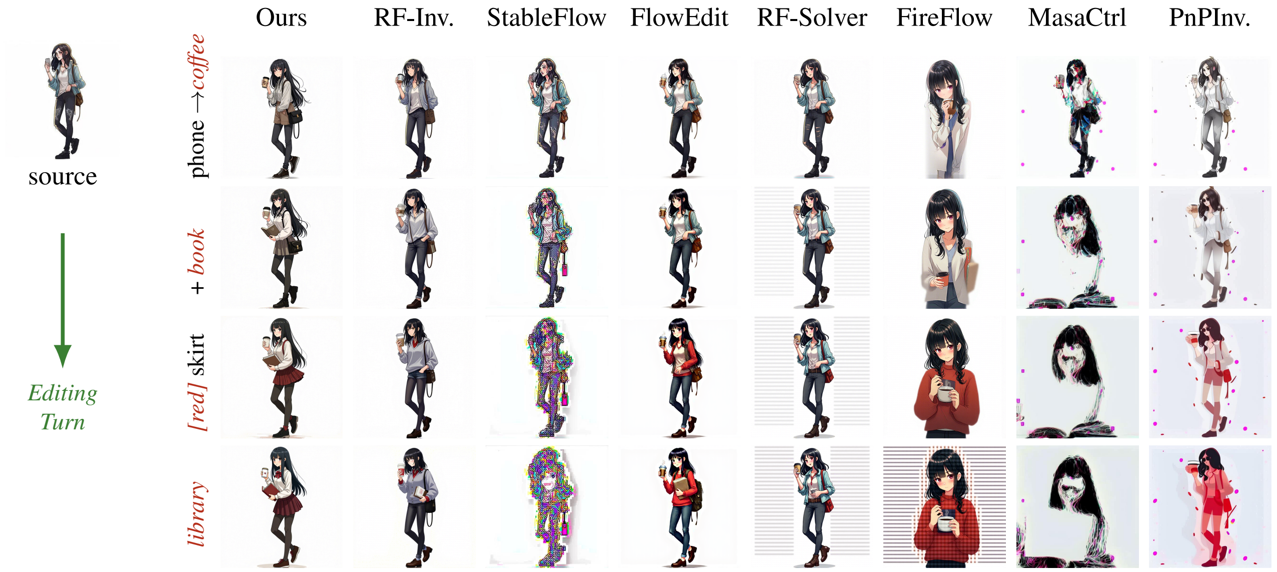


Figure 5. Quantitative results on artificial images show that our method successfully preserves the background while performing the editing.

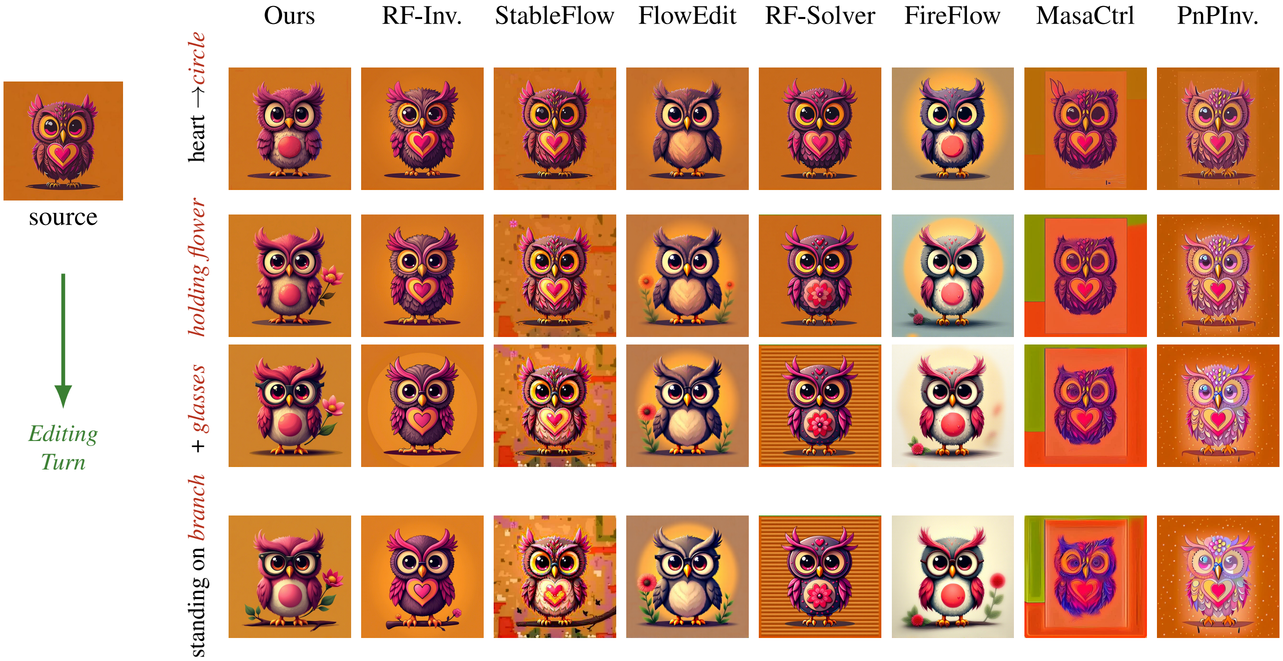


Figure 6. Quantitative results on artificial images show that our method successfully preserves the background while performing the editing.

References

[1] Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants, 2023. 2

[2] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. arXiv preprint arXiv:2411.14430, 2024. 2

[3] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In Proceedings of the IEEE/CVF international conference on computer vision, pages 22560–22570, 2023. 2

[4] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM transactions on Graphics (TOG), 42(4):1–10, 2023.

[5] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 5343–5353, 2024.

[6] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8795–8805, 2024. 2

[7] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. arXiv preprint arXiv:2210.11427, 2022. 2

[8] Xing Cui, Zekun Li, Peipei Li, Yibo Hu, Hailin Shi, and Zhaofeng He. CHATEDIT: Towards Multi-turn Interactive Facial Image Editing via Dialogue. https://arxiv.org/abs/2303.11108v3, 2023. 2

[9] Yingying Deng, Xiangyu He, Fan Tang, and Weiming Dong. $\hat{z}*$: Zero-shot style transfer via attention rearrangement. arXiv preprint arXiv:2311.16491, 2023. 2

[10] Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. FireFlow: Fast Inversion of Rectified Flow for Image Semantic Editing, 2024. 2

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis, 2024. 2

[12] Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6986–6996, 2024. 2

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626, 2022. 2

[14] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4775–4785, 2024. 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 2

[16] Nisha Huang, Fan Tang, Weiming Dong, Tong-Yee Lee, and Changsheng Xu. Region-aware diffusion for zero-shot text-driven image editing. arXiv preprint arXiv:2302.11797, 2023. 2

[17] KJ Joseph, Prateksha Udhayanan, Tripti Shukla, Aishwarya Agarwal, Srikrishna Karanam, Koustava Goswami, and Balaji Vasan Srinivasan. Iterative multi-granular image editing using diffusion models. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 8107–8116, 2024. 2

[18] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. arXiv preprint arXiv:2310.01506, 2023. 2

[19] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1931–1941, 2023. 2

[20] Huayang Li, Siheng Li, Deng Cai, Longyue Wang, Lemao Liu, Taro Watanabe, Yujiu Yang, and Shuming Shi. TextBind: Multi-turn Interleaved Multimodal Instruction-following in the Wild, 2024. 2

[21] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, 2023. 2

[22] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7817–7826, 2024. 2

[23] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, 2022. 2

[24] Zerun Liu, Fan Zhang, Jingxuan He, Jin Wang, Zhangye Wang, and Lechao Cheng. Text-guided mask-free local image retouching. In 2023 IEEE International Conference on Multimedia and Expo (ICME), pages 2783–2788. IEEE, 2023. 2

[25] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. arXiv preprint arXiv:2305.16807, 2023. 2

[26] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6038–6047, 2023. 2

[27] Jisu Nam, Heesu Kim, DongJae Lee, Siyoon Jin, Seungryong Kim, and Seunggyu Chang. Dreammatcher: appearance matching self-attention for semantically-consistent text-to-image personalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8100–8110, 2024. 2

[28] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International conference on machine learning, pages 8162–8171. PMLR, 2021. 2

[29] OpenAI. Gpt-4v(ision) system card, 2023. 2

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PmLR, 2021. 5

[31] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic Image Inversion and Editing using Rectified Stochastic Differential Equations, 2024. 2

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020. 2

[33] Nick Stracke, Stefan Andreas Baumann, Joshua Susskind, Miguel Angel Bautista, and Björn Ommer. Ctrloralter: Conditional loradapter for efficient 0-shot control and altering of t2i models. In European Conference on Computer Vision, pages 87–103. Springer, 2024. 2

[34] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1921–1930, 2023. 2

[35] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming Rectified Flow for Inversion and Editing, 2024. 2

[36] Ruichen Wang, Zekang Chen, Chen Chen, Jian Ma, Haonan Lu, and Xiaodong Lin. Compositional text-to-image synthesis with attention map control of diffusion models. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 5544–5552, 2024. 2

[37] Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2Img: Iterative Self-Refinement with GPT-4V(ision) for Automatic Image Design and Generation, 2023. 2

[38] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721, 2023. 2

[39] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF international conference on computer vision, pages 3836–3847, 2023. 2

[40] Yanbing Zhang, Mengping Yang, Qin Zhou, and Zhe Wang. Attention calibration for disentangled text-to-image personalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4764–4774, 2024. 2